

## BAB III

### METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

#### 3.1 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penelitian ini adalah dengan menggunakan SDLC. Hal pertama yang dilakukan adalah menganalisis kebutuhan melalui studi literatur dan pengumpulan data. Lalu, melakukan desain sistem dengan membuat diagram. Setelah itu dilakukan implementasi dengan melakukan pemrograman. Ketika sistem sudah selesai dibuat akan dilakukan pengujian sistem dan dievaluasi. Terakhir, akan dibuat naskah penelitian. Untuk detailnya dapat dijabarkan sebagai berikut:

a. Studi Literatur

Dilakukan dengan mempelajari sumber-sumber di internet seperti artikel, jurnal, buku yang berhubungan dengan penelitian ini untuk mendapatkan informasi dasar. Informasi yang dipelajari dalam penelitian ini adalah tingkat kesegaran kuning telur, metode CBIR, dan klasifikasi *Decision Tree* (CART).

b. Pengumpulan Data

Dilakukan dengan mengambil gambar kuning telur dengan menggunakan kamera *smartphone*. Pengambilan gambar dilakukan merujuk dengan penelitian “Egg Freshness Detection Based on Digital Image Technology” (Qiaohua Wang, 2009) dengan cara telur mentah dipecahkan sebanyak 10 butir dan diambil gambarnya perhari sampai telur menjadi busuk. Pengambilan gambar dilakukan di ruang terbuka pada antara jam 12 hingga jam 1 siang Waktu Indonesia Barat agar mendapat pencahayaan yang cukup. Jumlah gambar kuning telur yang terkumpul

dan dapat dipakai adalah 29 gambar. Gambar – gambar tersebut ditentukan kelasnya dengan merujuk pada penelitian “Development of Color-Based Extraction Methods for Nutritional Quality Grouping of Chicken Eggs” (Yanto, Madenda, & ETP, 2019). Penelitian tersebut membagi gambar kuning telur menjadi 3 , yaitu *Orange Yellow Egg* (level 11 - 15), *Ordinary Egg* (level 6 - 10) dan *Pale Egg* (level 1 - 5). Gambar kuning telur yang digunakan dalam penelitian hanya dibagi menjadi 2, yaitu *Orange Yellow* dan *Ordinary Egg*. Hal tersebut dikarenakan saat pengambilan gambar hanya terdapat kuning telur yang berada pada kedua kelas tersebut. Untuk mempermudah dalam pembuatan sistem ini, *Orang Yellow* akan diberikan label OY dan *Ordinary Egg* akan diberikan label OE.

c. Desain Sistem

Dilakukan dengan membuat diagram, seperti *flowchart* dari sistem yang akan dibuat.

d. Pemrograman Sistem

Dilakukan dengan menulis kode-kode program menggunakan Jupyter Notebook versi 4.4.0 dan bahasa pemrograman Python 3.

e. Pengujian Sistem

Dilakukan dengan uji coba sistem yang sudah dibuat untuk mencari kesalahan-kesalahan yang terjadi saat pemrograman. Jika ada kesalahan, maka dilakukan perbaikan dan menjalankan kembali sistem yang sudah dibuat.

f. Evaluasi

Dilakukan dengan menggunakan GridSearchCV (*Cross Validation*) untuk dicari tingkat akurasi terbaik melalui parameter yang ada pada model klasifikasi *Decision Tree* pada hasil *retrieve image*.

g. Penulisan Naskah Penelitian

Dilakukan agar penelitian dapat terdokumentasi dengan baik dan dapat dijadikan referensi untuk penelitian yang akan mendatang.

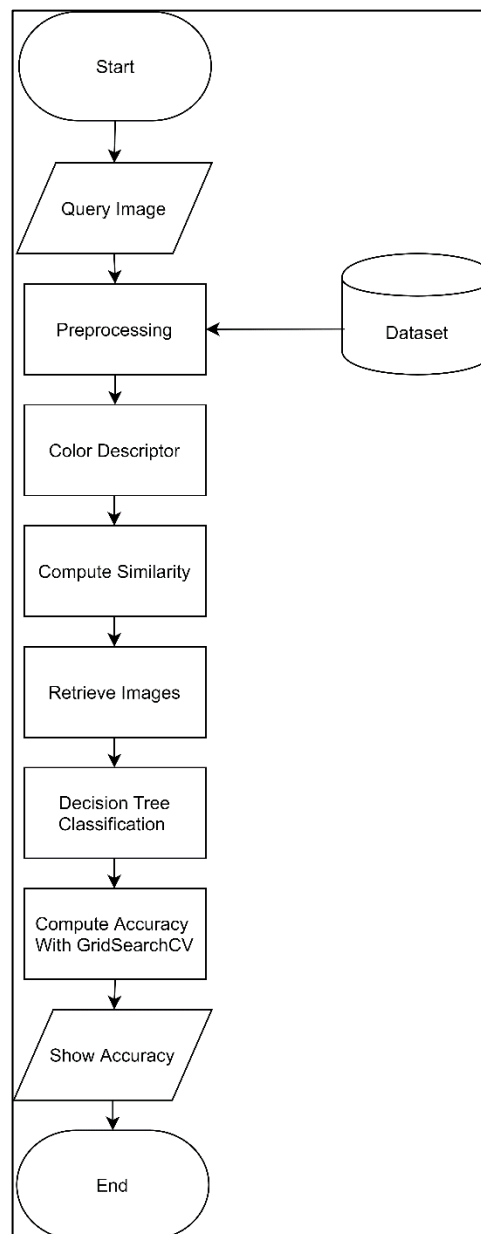
## 3.2 Perancangan Sistem

Perancangan sistem yang dibuat adalah menggunakan diagram *flowchart*.

### 3.2.1. *Flowchart*

*Flowchart* berikut merupakan alur kerja dari sistem yang akan dibuat

#### ***Flowchart* Sistem**



Gambar 3.1 *Flowchart* Sistem

### 1. *Query Image*

*Query Image* menjadi *input* dalam sistem ini dan diambil dari salah satu gambar di dataset.

### 2. *Preprocessing*

Tahap *Preprocessing* dilakukan dengan cara *cropping* dan *resizing images* pada *query image* dan dataset dengan menggunakan aplikasi Adobe Photoshop. Gambar di-*crop* dengan mengambil bagian kuning telur dengan ukuran 800 x 800 piksel. Ukuran tersebut diambil karena disesuaikan dari seluruh gambar yang masih mentah (*raw image*) dan dicari ukuran *cropping* yang dirasa cocok.

### 3. Color Descriptor

Tahap *Color Descriptor* merupakan tahap untuk mengambil informasi warna dari *query image* dan juga dataset. Pengambilan informasi warna pertama kali dilakukan dengan cara gambar diubah formatnya dari RGB menjadi HSV. HSV merupakan singkatan dari *Hue*, *Saturation* dan *Values*. *Hue* adalah warna yang dilihat oleh manusia, *Saturation* adalah tingkat intensitas warna, dan *Value* adalah tingkat kecerahan warna. Format warna HSV paling mendekati dengan penglihatan manusia terhadap warna. Setelah itu, dilakukan pembagian gambar menjadi 5 bagian, yaitu kiri atas, kanan atas, kanan bawah, kiri bawah, dan tengah gambar. Teknik pembagian tersebut dilakukan merujuk dari *website* pyimagesearch.com (Brock, 2014) yang menyatakan dalam ilmu fotografi, subjek atau objek yang akan diambil gambarnya biasanya diletakan ditengah sehingga ada

pembagian ditengah gambar yang ukurannya lebih besar dibanding 4 bagian lainnya dan teknik tersebut dapat bekerja dengan baik. Setiap bagian akan dihitung *histogram color*-nya dengan ukuran 6 x 3 x 3 bins. Ukuran tersebut dipilih karena cocok untuk digunakan pada metode pembagian gambar menjadi 5 bagian ini yang merujuk pada buku “Scalable Fuzzy Algorithms for Data Management and Analysis Methods and Designs” (Laurent & Lesot, 2010).

#### 4. *Compute Similarity*

Perhitungan kesamaan atau *similarity* dilakukan dengan perhitungan jarak menggunakan *Euclidean Distances*.

#### 5. *Retrieve Images*

Tahap *Retrieve Images* dilakukan dengan mencari jarak antara *query image* dengan seluruh gambar yang ada pada dataset yang memiliki nilai mendekati 0.

#### 6. *Decision Tree Classification*

Klasifikasi *Decision Tree* dilakukan dengan membagi data *retrieve images* menjadi 70% untuk *training* dan 30% untuk *testing*. *Node* dari *Decision Tree* adalah 270 data bertipe float yang didapatkan dari jumlah bins yang dipakai pada fitur ekstraksi warna. *Target Class* pada *Decision Tree* adalah OY dan OE. Untuk membuat model klasifikasi ini menggunakan perhitungan *gini index*. Berikut terdapat contoh gambar dengan data bins dan *class*-nya (bukan data yang digunakan dalam

penelitian ini dikarenakan jumlah data yang cukup banyak dan rumit) serta cara perhitungan *gini index*.

Tabel 3.1 Contoh Data Bins Gambar

Gambar	Bin 1	Bin 2	Bin 3	Target Class
1	0.2	0.5	0.4	OE
2	0.5	0.4	0.4	OY
3	0.2	0.5	0.2	OE
4	0.2	0.4	0.2	OY

- Bin 1

$$Gini (Bin 1 = 0.2) = 1 - (2/3)^2 - (1/3)^2 = 0.44$$

$$Gini (Bin 1 = 0.5) = 1 - (1/1)^2 = 0$$

$$Gini (Bin 1) = (3/4) \times 0.44 + (1/4) \times 0 = 0.33$$

- Bin 2

$$Gini (Bin 2 = 0.5) = 1 - (2/2)^2 = 0$$

$$Gini (Bin 2 = 0.4) = 1 - (2/2)^2 = 0$$

$$Gini (Bin 2) = (2/4) \times 0 + (2/4) \times 0 = 0$$

- Bin 3

$$Gini (Bin 3 = 0.4) = 1 - (1/2)^2 - (1/2)^2 = 0.5$$

$$Gini (Bin 3 = 0.2) = 1 - (1/2)^2 - (1/2)^2 = 0.5$$

$$Gini (Bin 3) = (2/4) \times 0.5 + (2/4) \times 0.5 = 0.5$$

Dari perhitungan *gini index* diatas akan diambil hasil yang paling kecil, yaitu Bin 2. Bin 2 akan menjadi Root Node. Selanjutnya akan dihitung *gini index* kembali untuk dicari Internal Node dengan perhitungan dibawah ini.

- Bin 2 (0.5) with Bin 1

$$Gini (\text{Bin } 2 = 0.5, \text{Bin } 1 = 0.2) = 1 - (2/2)^2 = 0$$

$$Gini (\text{Bin } 2 (0.5) \text{ with Bin } 1) = (2/2) \times 0 = 0$$

- Bin 2 (0.5) with Bin 3

$$Gini (\text{Bin } 2 = 0.5, \text{Bin } 3 = 0.4) = 1 - (1/1)^2 = 0$$

$$Gini (\text{Bin } 2 = 0.5, \text{Bin } 3 = 0.2) = 1 - (1/1)^2 = 0$$

$$Gini (\text{Bin } 2 (0.5) \text{ with Bin } 3) = (1/2) \times 0 + (1/2) \times 0 = 0$$

Perhitungan *Gini Index* Bin 2 (0.5) menghasilkan nilai 0 pada Bin 1 dan Bin 3 sehingga dapat dipilih salah satu untuk dijadikan *node* selanjutnya. Untuk perhitungan selanjutnya dapat dilihat dibawah ini.

- Bin 2 (0.4) with Bin 1

$$Gini (\text{Bin } 2 = 0.4, \text{Bin } 1 = 0.5) = 1 - (1/2)^2 = 0$$

$$Gini (\text{Bin } 2 = 0.4, \text{Bin } 1 = 0.2) = 1 - (1/2)^2 = 0$$

$$Gini (\text{Bin } 2 (0.4) \text{ with Bin } 1) = (1/2) \times 0 + (1/2) \times 0 = 0$$

- Bin 2 (0.4) with Bin 3

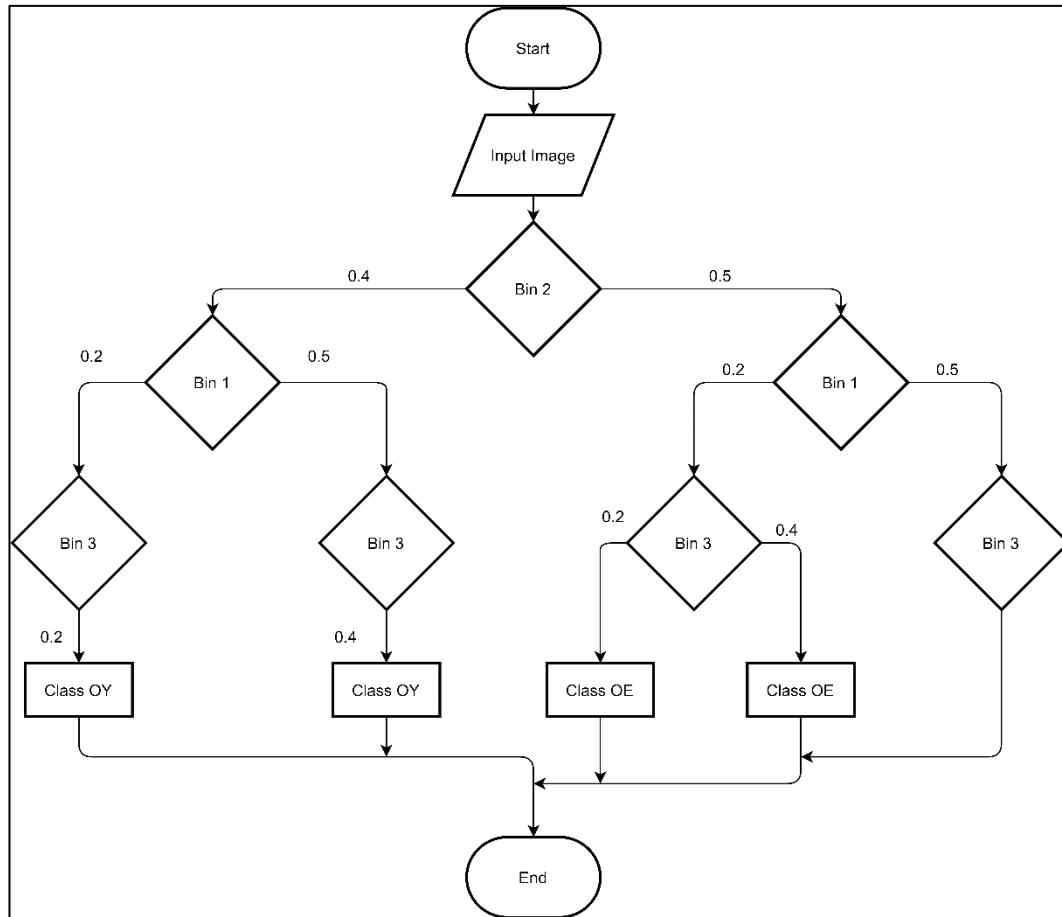
$$Gini (\text{Bin } 2 = 0.4, \text{Bin } 3 = 0.4) = 1 - (1/2)^2 = 0$$

$$Gini (\text{Bin } 2 = 0.4, \text{Bin } 3 = 0.2) = 1 - (1/2)^2 = 0$$

$$Gini (\text{Bin } 2 (0.4) \text{ with Bin } 3) = (1/2) \times 0 + (1/2) \times 0 = 0$$

Perhitungan *Gini Index* Bin 2 (0.4) menghasilkan nilai 0 pada Bin 1 dan Bin 3 sehingga dapat dipilih salah satu untuk dijadikan *node* selanjutnya. Dari perhitungan tersebut dapat digambarkan dengan *flowchart* dibawah ini.





Gambar 3.2 *Flowchart Decision Tree*

## 7. *Compute Accuracy With GridSearchCV*

Penggunaan GridSearchCV untuk melakukan *Cross Validation* dan mencari parameter terbaik dari model *Decision Tree* yang dibuat. *Cross Validation* yang dilakukan dengan menggunakan *K-fold Validation* dengan K sebanyak 7 folds. Angka tersebut diambil dikarena hasil *retrieve images* yang digunakan pada penelitian ini adalah 20, sehingga ketika dibagi *training* dan *testing* jumlah data *training* adalah 14. Jumlah data *training* dengan sebanyak itu akan dibagi kedalam *K-folds* dengan isi data tiap *folds* sama sehingga dipakai  $K = 7$ . Untuk parameter yang akan dicari adalah

*criterion* (kriteria *splitting*), *max\_depth* (jumlah kedalaman pohon), *max\_features* (jumlah maksimal *features*), *min\_samples\_split* (jumlah data maksimal dalam *internal node*) dan *min\_samples\_leaf* (jumlah data maksimal dalam daun). Dengan begitu, GridSearchCV dapat mencari akurasi yang terbaik.

#### 8. *Show Accuracy*

Tahap ini merupakan *output* dari sistem yang dibuat, yaitu akurasi yang terbaik serta parameter yang terbaik.